

Solving Verbal Comprehension Questions in IQ Test by Knowledge-Powered Word Embedding^{*}

Huazheng Wang
Dept. of Computer Science
University of Science and
Technology of China
wanghzh@mail.ustc.edu.cn

Bin Gao
Microsoft Research
13F, Bldg 2, No. 5, Danling St
Beijing, 100080, P. R. China
bingao@microsoft.com

Jiang Bian
Microsoft Research
13F, Bldg 2, No. 5, Danling St
Beijing, 100080, P. R. China
jibian@microsoft.com

Fei Tian
Dept. of Computer Science
University of Science and
Technology of China
tianfei@mail.ustc.edu.cn

Tie-Yan Liu
Microsoft Research
13F, Bldg 2, No. 5, Danling St
Beijing, 100080, P. R. China
tyliu@microsoft.com

ABSTRACT

Intelligence Quotient (IQ) Test is a set of standardized questions designed to evaluate human intelligence. Verbal comprehension questions appear very frequently in IQ tests, which measure human's verbal ability including the understanding of the words with multiple senses, the synonyms and antonyms, and the analogies among words. In this work, we explore whether such tests can be solved automatically by artificial intelligence technologies, especially the deep learning technologies that are recently developed and successfully applied in a number of fields. However, we found that the task was quite challenging, and simply applying existing technologies (e.g., word embedding) could not achieve a good performance, mainly due to the multiple senses of words and the complex relations among words. To tackle this challenge, we propose a novel framework consisting of three components. First, we build a classifier to recognize the specific type of a verbal question (e.g., analogy, classification, synonym, or antonym). Second, we obtain distributed representations of words and relations by leveraging a novel word embedding method that considers the multi-sense nature of words and the relational knowledge among words (or their senses) contained in dictionaries. Third, for each specific type of questions, we propose a simple yet effective solver based on the obtained distributed word representations and relation representations. According to our experimental results, our proposed framework can not only outperform existing methods for solving verbal comprehension questions but also exceed the average performance of the Amazon Mechanical Turk workers involved in the ex-

periments. The results are highly encouraging, indicating that with appropriate uses of the deep learning technologies, we could be a further small step closer to the human intelligence.

Keywords

Machine learning, deep learning, word embedding, IQ test.

1. INTRODUCTION

Intelligence is the ability or capacity that enables the individuals to deal with real situations and profit intellectually from sensory experience. A test of intelligence is designed to formally study the success of an individual in adapting to a specific situation under certain conditions. The most famous test is the Intelligence Quotient (IQ) test, which was first proposed about 100 years ago [29]. Common IQ tests measure various types of abilities such as verbal, mathematical, spatial and reasoning skills. These tests have been widely used in the study of psychology, education, career development, etc. Although a high IQ might not be a necessary condition for a successful life, people would still tag the individual with high IQ by labels of *smart* or *clever*.

Artificial Intelligence (AI) is the human-like intelligence exhibited by machines or software, which was first activated by the famous Turing test [32] and has attracted a lot of people to study afterwards. The main task of AI is to study and design intelligent agents that perceive the environment and take actions to maximize the chances of success. With the fast development of artificial intelligence, agents have been invented to fulfill many interesting and challenging tasks like face recognition, speech recognition, handwriting recognition, robot soccer, question answering, chess, and natural language processing. However, as far as we know, there are very limited studies of developing an agent to solve IQ tests yet, which in some sense is more challenging, since even common human beings could not always succeed in the tests. Considering that IQ tests have been widely considered as a measure of *intelligence*, we think it is worth making further investigations whether we can develop an agent that can beat human on solving IQ tests.

The commonly used IQ tests contain a few types of questions, among which a significant proportion (around 40%)

^{*}This work was performed when the first and the fourth authors were interns at Microsoft Research Asia.

are verbal questions [8]. The recent progress on deep learning for natural language processing (NLP), such as word embedding technologies, has advanced the ability of machines (or AI agents) to understand the meaning of words and the relations among words. **This inspires us to solve the verbal questions in IQ tests by leveraging the word embedding technologies.** However, our attempts show that a straightforward application of the word embedding technologies could not result in satisfactory performances. This is actually understandable. Standard word embedding technologies learn one embedding vector for each word based on the co-occurrence information in a text corpus. However, verbal comprehension questions in IQ test usually consider the multiple senses of a word (and often focus on the rare senses), and the complex relations among (polysemous) words. This has clearly exceeded the capability of standard word embedding technologies.

To tackle the aforementioned challenge, we propose a novel framework which consists of three components.

First, we build a classifier to recognize the specific type of verbal questions. According to previous studies [8], verbal questions usually include sub-types like analogy, classification, synonym, and antonym. For different types of questions, different kinds of relationships need to be considered and the solvers could have different forms. Therefore, with an effective question type classifier, we may solve the questions in a divide-and-conquer manner and achieve high accuracy.

Second, we obtain distributed representations of words and relations by leveraging a novel word embedding method that considers the multi-sense nature of words and the relational knowledge among words (or their senses) contained in dictionaries. In particular, for each polysemous word, we retrieve its number of senses from a dictionary, and conduct clustering on all its context windows in the corpus. Then we attach the example sentences for every sense in the dictionary to the clusters, such that we can tag the polysemous word in each context window with a specific word sense. On top of this, instead of learning one embedding vector for each word, we are able to learn one vector for each pair of word-sense. Furthermore, in addition to learning the embedding vectors for words, we also learn the embedding vectors for relations (e.g., synonym and antonym) at the same time, by incorporating relational knowledge into the objective function of the word embedding learning algorithm. That is, the learning of word-sense representations and relation representations interacts with each other, to effectively incorporate the relational knowledge obtained from dictionaries.

Third, for each specific type of questions, we propose a simple yet effective solver based on the obtained distributed word-sense representations and relation representations. For example, for analogy questions, we find the answer by minimizing the distance between word-sense pairs in the question and the word-sense pairs in the candidate answers; for antonym questions, we calculate the offset between the representation of the question word of every sense and the representation of each candidate word with every of their possible senses, and then we find the answer by minimizing the distance between the antonym relation representation and the above offsets.

We have conducted experiments using a combined IQ test set to test the performance of our proposed framework. The experimental results show that our method can significantly

outperform several baseline methods for verbal comprehension questions in IQ test. We further deliver the questions in the test set to human beings through Amazon Mechanical Turk¹. To our surprise, the average performance of the human beings is even a little lower than that of our proposed method. This is highly encouraging, which somehow shows that with appropriate uses of the deep learning technologies, we could be a further step closer to the true human intelligence.

2. RELATED WORK

2.1 IQ Test

Intelligence Quotient Test was proposed by William Stern as a scoring method for human intelligence about a century ago [29]. Usually, such tests contain tens of questions for human to complete within a limited time, and then an IQ score is calculated according to the correctness of the answers and several other factors like the age of the human, the time to complete the test, and the behaviors the human perform during the test. Currently, in the mainstream IQ tests, the median raw score of the norming sample is defined as 100 IQ points and the scores each standard deviation up or down are defined as 15 IQ points greater or less. By this definition, approximately 95% of the population scores an IQ between 70 and 130, which is within two standard deviations of the mean.

Common IQ tests mainly contain three categories of questions [8]: verbal comprehensive questions, mathematical questions, and logic questions. Verbal questions include several types like analogy, classification, synonym, and antonym, which will be introduced with more details in Section 3. Mathematical questions include several types like algebra, number sequence, and math logic. For example, the task of number sequence problems is to extrapolate some finite sequences of integers. Logic questions often appear in pictorial matrices, the task of which is to identify the missing element that completes the pattern of a pictorial matrix.

There has been very few efforts to develop automatic methods to solve IQ tests. Sanghi and Dowe [26] presented a fairly elementary WWW-based computer program that can solve a variety of IQ tests, regularly obtains a score close to the purported average human score of 100. Strannegard et al. [30] proposed an anthropomorphic method for number sequence solvers that targets performance on the level of human role models, and they also presented a method [30] for solving progressive matrix problems in logic questions. Recently, Kushmany et al. [19] proposed an approach for automatically learning to solve algebra word problems, Seo et al. [27] proposed a method to automatically solve geometry problems, and Hosseini et al. [16] proposed an approach to learning to solve simple arithmetic word problems.

Besides the above efforts, there are also some debates on whether IQ tests are appropriate for evaluating the intelligence of machines or measuring the progress in AI [13], because some IQ test questions might be easily hacked and then correctly answered by some simple tricks. However, we think that these tricks are not principle methods and they cannot be generalized to more novel questions. We would like to do more exploration on using automatical algorithms to solve the IQ test problems.

¹<http://www.mturk.com/>

2.2 Deep Learning for Text Mining

Building distributed word representations [2], a.k.a. word embeddings, has attracted increasing attention in the area of machine learning. Different with conventional one-hot representations of words or distributional word representations based on co-occurrence matrix between words such as LSA [14] and LDA [5], distributed word representations are usually low-dimensional dense vectors trained with neural networks by maximizing the likelihood of a text corpus. Recently, to show its effectiveness in a variety of text mining tasks, a series of works applied deep learning techniques to learn high-quality word representations. For example, Collobert et al. [10, 11] proposed a neural network that can learn a unified word representations suited for several NLP tasks simultaneously. Furthermore, Mikolov et al. proposed efficient neural network models for learning word representations, i.e. *word2vec* [21]. Under the assumption that similar words yield similar context, the *word2vec* model maximizes the log likelihood of each word given its context words within a sliding window. The learned word representations show that they can indicate both syntactic and semantic regularities among words.

Nevertheless, since the above works learn word representations mainly based on the word co-occurrence information, it is quite difficult to obtain high quality embeddings for those words with very little context information; on the other hand, large amount of noisy or biased context could give rise to ineffective word embeddings either. Therefore, it is necessary to introduce extra knowledge into the learning process to regularize the quality of word embedding. Some efforts have paid attention to learn word embedding in order to address knowledge base completion and enhancement [7, 28, 34]; however, they did not investigate the other side of the coin, i.e. leveraging knowledge to enhance word representations. Recently, there have been some early attempts on this direction. For example, Luong et al. [20] proposed a neural network model to learn word representations by leveraging morphological knowledge on words. Yu et al. [37] proposed a new learning objective that incorporates both a neural language model objective and a semantic prior knowledge objective to learn improved word representations. Bian et al. [3] recently proposed to leverage morphological, syntactic, and semantic knowledge to advance the learning of word embeddings. Particularly, they explored these types of knowledge to define new basis for word representations, provide additional input information, and serve as auxiliary supervision in the learning process.

Moreover, all the above models assume that one word has only one embedding no matter whether the word is polysemous or monosemous, which might bring some confusion for the polysemous words. To solve the problem, Huang et al. [17] leveraged the global context information to train an initial word embedding and then proposed a clustering based method to produce multi-sense word embeddings for polysemous words. Recently, Tian et al. [31] proposed to model word polysemy from a probabilistic perspective and integrate it with the *word2vec* model. However, these models do not leverage any extra knowledge (e.g., relational knowledge) to enhance word representations.

In contrast to all the aforementioned works, in this paper, we present a novel method that can produce multi-sense word embeddings powered by relational knowledge, which is more effective to solve the verbal comprehension questions.

3. VERBAL QUESTIONS IN IQ TEST

In common IQ tests, a large proportion of questions are verbal comprehension questions, which play an important role in deciding the final IQ scores. For example, in Wechsler Adult Intelligence Scale [33], which is among the most famous IQ test systems, the full-scale IQ is calculated from two IQ scores: Verbal IQ and Performance IQ, and around 38% questions in a typical test are verbal comprehension questions. In another popular system named Woodcock-Johnson Tests of Cognitive Abilities [35], the final IQ score is derived from three tests including the Verbal Comprehension Test. Verbal questions can test not only the verbal ability (e.g., understanding polysemy of a word), but also the reasoning ability and induction ability of an individual. According to previous studies [8], verbal questions mainly have the following types: analogy, classification, synonym, and antonym, which are elaborated in detailed as below.

3.1 Analogy-I

Analogy-I questions usually take the form “*A* is to *B* as *C* is to ?”. One needs to choose a word *D* from a given list of candidate words to form an analogical relation between pair (*A*, *B*) and pair (*C*, *D*). Such questions test the ability of identifying an implicit relation from word pair (*A*, *B*) and apply it to compose word pair (*C*, *D*). Here is an example.

EXAMPLE 1. *Isotherm is to temperature as isobar is to?*
(i) *atmosphere*, (ii) *wind*, (iii) *pressure*, (iv) *latitude*, (v) *current*.

The correct answer is *pressure*, because an isotherm is an isogram connecting points with the same temperature, while an isobar is an isogram connecting points with the same pressure. Note that the Analogy-I questions are also used as a major evaluation task in the *word2vec* models [21].

3.2 Analogy-II

Analogy-II questions require two words to be identified from two given lists in order to form an analogical relation like “*A* is to ? as *C* is to ?”. Here is an example.

EXAMPLE 2. *Identify two words (one from each set of brackets) that form a connection (analogy) when paired with the words in capitals: CHAPTER (book, verse, read), ACT (stage, audience, play).*

The correct answer is *book*, *play*, because a book is composed by several chapters and a play is composed by several acts. Such questions are a bit more difficult than the Analogy-I questions since the analogical relation cannot be observed directly from the questions, but need to be searched in the word pair combinations from the candidate answers.

3.3 Classification

Classification questions require one to identify the word that is different (or dissimilar) from others in a given word list. Such questions are also known as *Odd-One-Out*, which have been studied in [24]. Classification questions test the ability of summarizing the majority sense of the words and identifying the outlier. Here is a typical example.

EXAMPLE 3. *Which is the odd one out?* (i) *calm*, (ii) *quiet*, (iii) *relaxed*, (iv) *serene*, (v) *unruffled*.

The correct answer is *quiet*, which means the absence of sound, while the other words all have the similar meaning to *calm*.

3.4 Synonym

Synonym questions require one to pick one word out of a list of words such that it has the closest meaning to a given word. Synonym questions test the ability of identifying all senses of the candidate words and selecting the correct sense that can form a synonymous relation to the given word. Here is a typical example.

EXAMPLE 4. Which word is closest to *IRRATIONAL*?
(i) *intransigent*, (ii) *irredeemable*, (iii) *unsafe*, (iv) *lost*, (v) *nonsensical*.

The correct answer is *nonsensical*. The word *irrational* has multiple senses, including (i) *without power to reason* which is used in the context of psychology, (ii) *unreasonable* which is also used in the context of psychology, and (iii) *real number that cannot be expressed as the quotient of two integers* which is used in the context of mathematics. In this question, the closest word is *nonsensical*, which is synonymous to the second sense of *irrational*.

3.5 Antonym

Antonym questions require one to pick one word out of a list of words such that it has the opposite meaning to a given word. Antonym questions test the ability of identifying all senses of the candidate words and selecting the correct sense that can form an antonymous relation to the given word. Here is a typical example.

EXAMPLE 5. Which word is most opposite to *MUSICAL*?
(i) *discordant*, (ii) *loud*, (iii) *lyrical*, (iv) *verbal*, (v) *euphony*.

The correct answer is *discordant*. *Musical* here means pleased by harmonious melody, while *discordant* means *lacking in harmony*.

From the above explanations, we can see that for different types of questions, one had better consider different kinds of relationships and the solvers should have different forms. Therefore, divide-and-conquer could be a good strategy, and we will design our framework on this basis.

4. SOLVING VERBAL QUESTIONS

In this section, we introduce our proposed framework to solve the verbal questions, which consists of the following three components.

4.1 Classification of Question Types

The first component of the framework is a question classifier, which identifies different types of verbal questions. Since different types of questions usually have their unique ways of expressions, the classification task is relatively easy, and we therefore take a simple approach to fulfill the task. Specifically, we regard each verbal question as a short document and use the TF-IDF [1] feature to build its representation. Then we train an SVM [12] classifier with linear kernel on a portion of labeled question data, and apply it to other questions. The question labels include Analogy-I, Analogy-II, Classification, Synonym, and Antonym. We use the *one-vs-rest* training strategy [4] to obtain a linear SVM classifier for each question type.

4.2 Embedding of Word-Senses and Relations

The second component of our framework leverages deep learning technologies to learn distributed representations for words (i.e. word embedding). Note that in the context of verbal question answering, we have some specific requirements on this learning process. Verbal questions in IQ test usually consider the multiple senses of a word (and focus on the rare senses), and the complex relations among (polysemous) words. Figure 1 shows an example on the multi-sense of words and the relations among word senses. We can see that *irrational* have three senses. Its first sense has an antonym relation with the second sense of *rational*, while its second sense has a synonym relation with *nonsensical* and an antonym relation with the first sense of *rational*.

The above challenge has exceeded the capability of standard word embedding technologies. To address this problem, we propose a novel approach that considers the multi-sense nature of words and integrate the relational knowledge among words (or their senses) into the learning process. In particular, our approach consists of two steps. The first step aims at labeling a word in the text corpus with its specific sense, and the second step employs both the labeled text corpus and the relational knowledge contained in dictionaries to simultaneously learn embeddings for both word-sense pairs and relations.

4.2.1 Multi-Sense Identification

While word embedding has shown its success in many text mining applications, one common limitation of existing studies is the assumption of single-sense representation. In practice, many words have multiple senses, which can be wildly different from each other. Thus, learning one single embedding vector for each word simply cannot capture the different senses of polysemous words. Several previous studies [25, 17, 31] have proposed using multiple representations to capture the different senses of a word. In this framework, we present a way, similar to [17], of using **pre-learned single-sense embeddings to represent each context window**, which can then be clustered to perform word sense discrimination. Beyond the method in [17], we also take advantages of additional knowledge in online dictionaries to regularize the word sense discrimination.

First, we learn a single-sense word embedding by using the skip-gram method in *word2vec* [21] (see Figure 2). In particular, a sliding window is employed on the input text stream to generate the training samples. In each sliding window, the model tries to use the central word as input to predict the surrounding words. Given a sequence of training text stream $w_1, w_2, w_3, \dots, w_K$, the objective of the skip-gram model is to maximize the following average log probability:

$$L = \frac{1}{K} \sum_{k=1}^K \sum_{-N \leq j \leq N, j \neq 0} \log p(w_{k+j} | w_k), \quad (1)$$

where w_k is the central word, w_{k+j} is a surrounding word, K is the length of the training text stream, and N indicates the context window size is $2N + 1$. The conditional probability $p(w_{k+j} | w_k)$ is defined in the following *softmax* function:

$$p(w_{k+j} | w_k) = \frac{\exp(v'_{w_{k+j}} v_{w_k}^T)}{\sum_{w=1}^V \exp(v'_w v_{w_k}^T)}, \quad (2)$$

where v_w and v'_w are the input and output latent variables,

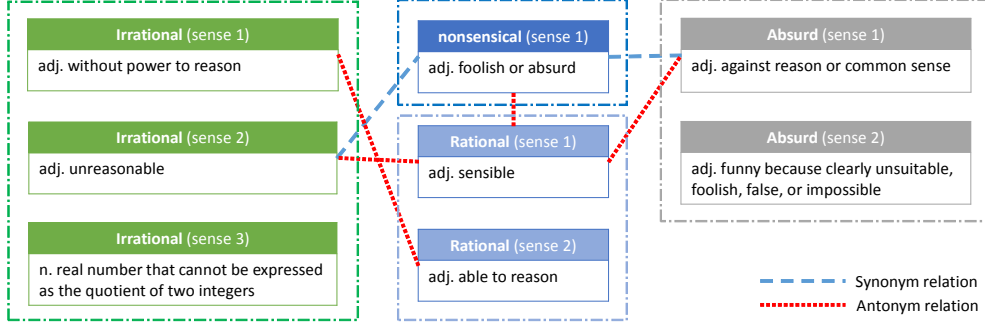


Figure 1: An example on the multi-sense of words and the relations between word senses.

i.e. the input and output representation vectors of w , and V is the vocabulary size.

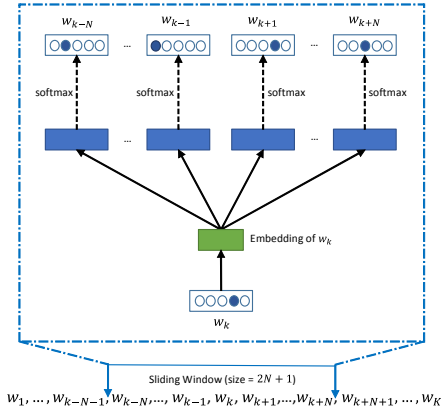


Figure 2: The skip-gram model.

Second, we gather the context windows of all occurrences of a word used in the skip-gram model, and represent each context by a weighted average of the pre-learned embedding vectors of the context words. We use TF-IDF to define the weighting function, where we regard each context window of the word as a short document to calculate the document frequency. Specifically, for a word w_0 , each of its context window can be denoted by $(w_{-N}, \dots, w_0, \dots, w_N)$. Then we represent the window by calculating the weighted average of the pre-learned embedding vectors of the context words as below,

$$\xi = \frac{1}{2N} \sum_{i=-N, i \neq 0}^N g_{w_i} v_{w_i}, \quad (3)$$

where g_{w_i} is the TF-IDF score of w_i , and v_{w_i} is the pre-learned embedding vector of w_i . After that, for each word, we use spherical k -means to cluster all its context representations, where cluster number k is set as the number of senses of this word in the online dictionary.

Third, we match each cluster to the corresponding sense in the dictionary. On one hand, we represent each cluster by the average embedding vector of all those context windows included in the cluster. For example, suppose word w_0 has k senses and thus it has k clusters of context windows, we denote the average embedding vectors for these clusters as ξ_1, \dots, ξ_k . On the other hand, since the online dictionary uses some descriptions and example sentences to interpret each word sense, we can represent each word sense

by the average embedding of those words including its description words and the words in the corresponding example sentences. Here, we assume the representation vectors (based on the online dictionary) for the k senses of w_0 are ζ_1, \dots, ζ_k . After that, we consecutively match each cluster to its closest word sense in terms of the distance computed in the word embedding space, i.e.,

$$(\bar{\xi}_{i'}, \zeta_{j'}) = \underset{i, j=1, \dots, k}{\operatorname{argmin}} d(\bar{\xi}_i, \zeta_j), \quad (4)$$

where $d(\cdot, \cdot)$ calculates the Euclidean distance and $(\bar{\xi}_{i'}, \zeta_{j'})$ is the first matched pair of window cluster and word sense. Here, we simply take a greedy strategy. That is, we remove $\bar{\xi}_{i'}$ and $\zeta_{j'}$ from the cluster vector set and the sense vector set, and recursively run (4) to find the next matched pair till all the pairs are found. Finally, each word occurrence in the corpus is relabeled by its associated word sense, which will be used to learn the embeddings for word-sense pairs in Section 4.2.2.

4.2.2 Co-Learning Word-Sense Pair Representations and Relation Representations

After relabeling the text corpus, different occurrences of a polysemous word may correspond to its different senses, or more accurately word-sense pairs. We then learn the embeddings for word-sense pairs and relations (obtained from dictionaries, such as synonym and antonym) simultaneously, by integrating relational knowledge into the objective function of the word embedding learning model like skip-gram.

Inspired by some recent work on multi-relation model [6, 36] that builds relationships between entities by interpreting them as translations operating on the low-dimensional representations of the entities, we propose to use a function E_r as described below to capture the relational knowledge.

Specifically, the existing relational knowledge extracted from dictionaries, such as synonym, antonym, etc., can be naturally represented in the form of a triplet (*head, relation, tail*) (denoted by $(h_i, r, t_j) \in S$, where S is the set of relational knowledge), which consists of two word-sense pairs (i.e. word h with its i -th sense and word t with its j -th sense), $h, t \in W$ (W is the set of words) and a relationship $r \in R$ (R is the set of relationships). To learn the relation representations, we make an assumption that **relationships between words can be interpreted as translation operations and they can be represented by vectors**. The principle in this model is that if the relationship (h_i, r, t_j) exists, the representation of the word-sense pair t_j should be close to that of h_i plus the representation vector of the relationship r , i.e.

$h_i + r$; otherwise, $h_i + r$ should be far away from t_j . Note that this model learns word-sense pair representations and relation representations in a unified continuous embedding space.

According to the above principle, we define E_r as a margin-based regularization function over the set of relational knowledge S ,

$$E_r = \sum_{(h_i, r, t_j) \in S} \sum_{(h', r, t') \in S'} [\gamma + d(h_i + r, t_j) - d(h' + r, t')]_+.$$

Here $[X]_+$ denotes the positive part of X , $\gamma > 0$ is a margin hyperparameter, and $d(\cdot, \cdot)$ is the distance measure for the words in the embedding space. For simplicity, we again define $d(\cdot, \cdot)$ as the Euclidean distance. The set of corrupted triplets $S'_{(h, r, t)}$ is defined as:

$$S'_{(h, r, t)} = \{(h', r, t)\} \cup \{(h, r, t')\}, \quad (5)$$

which is constructed from S by replacing either the head word-sense pair or the tail word-sense pair by another randomly selected word with its randomly selected sense.

Note that the optimization process might trivially minimize E_r by simply increasing the norms of word-sense pair representations and relation representations. To avoid this problem, we use an additional constraint on the norms, which is a commonly-used trick in the literature [7]. However, instead of enforcing the L_2 -norm of the representations to 1 as used in [7], we adopt a soft norm constraint on the relation representations as below:

$$r_i = 2\sigma(x_i) - 1, \quad (6)$$

where $\sigma(\cdot)$ is the sigmoid function $\sigma(x_i) = 1/(1 + e^{-x_i})$, r_i is the i -th dimension of relation vector r , and x_i is a latent variable, which guarantees that every dimension of the relation representation vector is within the range $(-1, 1)$.

By combining the skip-gram objective function and the regularization function derived from relational knowledge, we get the following combined objective J_r that incorporates relational knowledge into the word-sense pair embedding calculation process,

$$J_r = \alpha E_r - L, \quad (7)$$

where α is the combination coefficient. Our goal is to minimize the combined objective J_r , which can be optimized using back propagation neural networks. Figure 3 shows the structure of the proposed model. By using this model, we can obtain the representations for both word-sense pairs and relations simultaneously.

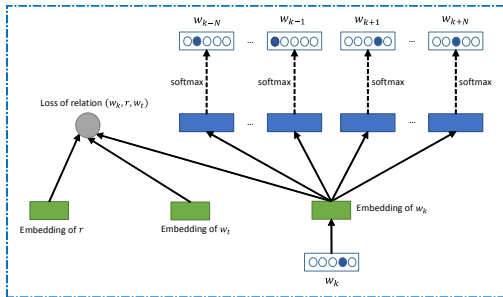


Figure 3: The structure of the proposed model.

4.3 Solver for Each Type of Questions

In this subsection, we define solvers for all types of verbal questions, by leveraging the embedding vectors for word-sense pairs and relations learned above.

4.3.1 Analogy-I

For the Analogy-I questions like “A is to B as C is to ?”, Mikolov et al. [21] showed that such analogical relations can be reflected by word vector offsets between each pair of words. For example, in *Man is to woman as king is to queen*, we have $v_{(woman)} - v_{(man)} \approx v_{(queen)} - v_{(king)}$. Inspired by this, we answer such questions by optimizing:

$$D = \underset{i_b, i_a, i_c, i_{d'}}{\operatorname{argmax}} \cos(v_{(B, i_b)} - v_{(A, i_a)} + v_{(C, i_c)}, v_{(D', i_{d'})}), \quad (8)$$

where T contains all the candidate answers, \cos means cosine similarity, and $i_b, i_a, i_c, i_{d'}$ are the indexes for the word senses of B, A, C, D' respectively. Finally D is selected as the answer.

4.3.2 Analogy-II

As the form of the Analogy-II questions is like “A is to ? as C is to ?” with two lists of candidate answers, we can apply an optimization method as below to select the best (B, D) pair,

$$\underset{i_{b'}, i_a, i_c, i_{d'}; B' \in T_1, D' \in T_2}{\operatorname{argmax}} \cos(v_{(B', i_{b'})} - v_{(A, i_a)} + v_{(C, i_c)}, v_{(D', i_{d'})}), \quad (9)$$

where T_1, T_2 are two lists of candidate words. Thus we get the answers B and D that can form an analogical relation between word pair (A, B) and word pair (C, D) under a certain specific word sense combination.

4.3.3 Classification

For the Classification questions, we leverage the property that words with similar co-occurrence information are distributed close to each other in the embedding space. As there is one word in the list that does not belong to others, it does not have similar co-occurrence information with other words in the training corpus, and thus this word should be far away from other words in the word embedding space.

According to the above discussion, we first calculate a group of mean vectors $m_{i_{w_1}, \dots, i_{w_N}}$ of all the candidate words with any possible word senses as below,

$$m_{i_{w_1}, \dots, i_{w_N}} = \frac{1}{N} \sum_{w_j \in T} v_{(w_j, i_{w_j})}, \quad (10)$$

where T is the set of candidate words, N is the capacity of T , w_j is a word in T ; $i_{w_j} (j = 1, \dots, N; i_{w_j} = 1, \dots, k_{w_j})$ is the index for the word senses of w_j , and $k_{w_j} (j = 1, \dots, N)$ is the number of word senses of w_j . Therefore, the number of the mean vectors is $M = \prod_{j=1}^N k_{w_j}$. As both N and k_{w_j} are very small, the computation cost is acceptable. Then, we choose the word with such a sense that it is the farthest away from one of the mean vectors in the embedding space as the answer, i.e.,

$$w = \underset{i_{w_j}; w_j \in T; l=1, \dots, M}{\operatorname{argmax}} d(v_{(w_j, i_{w_j})}, m_l). \quad (11)$$

4.3.4 Synonym

For the Synonym questions, we empirically explored two solvers. For the first solver, we also leverage the property

that words with similar co-occurrence information are located closely in the word embedding space. Therefore, given the question word w_q and the candidate words w_i , we can find the answer w by the following optimization problem.

$$w = \operatorname{argmin}_{i_{w_q}, i_{w_j}; w_j \in T} d(v_{(w_j, i_{w_j})}, v_{(w_q, i_{w_q})}), \quad (12)$$

where T is the set of candidate words. The second solver is based on the minimization objective of the translation distance between entities in the relational knowledge model (5). Specifically, we calculate the offset vector between the embedding of question word w_q and each word w_j in the candidate list. Then, we set the answer w as the candidate word with which the offset is the closest to the representation vector of the synonym relation r_s , i.e.,

$$w = \operatorname{argmin}_{i_{w_q}, i_{w_j}; w_j \in T} ||v_{(w_j, i_{w_j})} - v_{(w_q, i_{w_q})}|| - r_s|. \quad (13)$$

In practice, we found the second solver performs better (the results are listed in Section 5).²

4.3.5 Antonym

Similar to solving the Synonym questions, we explored two solvers for Antonym questions as well. That is, the first solver (14) is based on the small offset distance between semantically close words whereas the second solver (15) leverages the translation distance between two words' offset and the embedding vector of the antonym relation. One might doubt on the reasonableness of the first solver given that we aim to find an answer word with opposite meaning for the target word (i.e. antonym). We explain it here that since antonym and its original word have similar co-occurrence information, based on which the embedding vectors are derived, thus the embedding vectors of both words with antonym relation will still lie closely in the embedding space.

$$w = \operatorname{argmin}_{i_{w_q}, i_{w_j}; w_j \in T} d(v_{(w_j, i_{w_j})}, v_{(w_q, i_{w_q})}), \quad (14)$$

$$w = \operatorname{argmin}_{i_{w_q}, i_{w_j}; w_j \in T} ||v_{(w_j, i_{w_j})} - v_{(w_q, i_{w_q})}|| - r_a|, \quad (15)$$

where T is the set of candidate words and r_a is the representation vector of the antonym relation. Again we found that the second solver performs better. Similarly, for skip-gram, only the first solver is applied.

5. EXPERIMENTS

In this section, we conduct experiments to examine whether our proposed framework can achieve satisfying results on verbal comprehension questions.

5.1 Data Collection

5.1.1 Training Set for Word Embedding

In our experiments, we trained word embeddings on a publicly available text corpus named *wiki2014*³, which is a large text snapshot from Wikipedia. After being pre-processed

²For our baseline embedding modes skip-gram, since it does not assume the relation representations explicitly, we use the first solver for it.

³http://en.wikipedia.org/wiki/Wikipedia:Database_download

Table 1: Statistics of the verbal question test set.

Type of Questions	Number of questions
Analogy-I	50
Analogy-II	29
Classification	53
Synonym	51
Antonym	49
Total	232

by removing all the *html* meta-data and replacing the digit numbers by English words, the final training corpus contains totally more than 3.4 billion word tokens, and the number of unique words, i.e. the vocabulary size, is about 2 million.

5.1.2 IQ Test Set

According to our study, there is no online dataset specifically released for verbal comprehension questions, although there are many online IQ tests for users to play with. In addition, most of the online tests only calculate the final IQ scores but do not provide the correct answers. Therefore, we only use the online questions to train the verbal question classifier described in Section 4.1. Specifically, we manually collected and labeled 30 verbal questions from the online IQ test Websites⁴ for each of the five types (i.e. Analogy-I, Analogy-II, Classification, Synonym, and Antonym) and trained an *one-vs-rest* SVM classifier for each type. The total accuracy on the training set itself is 95.0%. The classifier was then applied in the test set below.

We collected a set of verbal comprehension questions associated with correct answers from the published IQ test books, such as [8, 9, 23, 18], and we used this collection as the test set to evaluate the effectiveness of our new framework. In total, this test set contains 232 questions with the corresponding answers.⁵ The statistics of each question type are listed in Table 1.

5.1.3 GRE Antonym Set

Graduate Record Examination (GRE) is a standardized test that is required in the admission process of most graduate schools in the United States, which can reflect students' advanced verbal ability. In our experiments, we applied our framework to a public GRE Antonym Dataset [22] containing 162 questions.

5.2 Compared Methods

In the following experiments, we compare our new relation knowledge powered model to several baselines.

Random Guess Model (RG). Random guess is the most straightforward way for an agent to solve questions. In our experiments, we used a random guess agent which would select an answer randomly regardless what the question was. To measure the performance of random guess, we ran each task for 5 times and calculated the average accuracy.

Human Performance (HP). Since IQ tests are designed to evaluate human intelligence, it is quite natural to leverage human performance as a baseline. To collect human answers on the test questions, we delivered them to human beings through Amazon Mechanical Turk, a crowdsourcing Internet marketplace that allows people to participate Human Intelligence Tasks. In our study, we published

⁴E.g., <http://wechsleradultintelligencescale.com/>

⁵It can be downloaded from <http://research.microsoft.com/en-us/um/beijing/events/DL-WSDM-2015/VerbalQuestions.zip>.

Table 2: Statistics of participants’ ages.

Age	Analogy-I	Analogy-II	Classification	Synonym	Antonym	GRE-Antonym
Under 18	0	0	0	0	0	0
18-29	63	67	79	87	91	60
30-39	72	60	47	60	50	58
40-60	56	66	64	47	51	72
Over 60	9	7	10	6	8	10
Overall	200	200	200	200	200	200

Table 3: Statistics of participants’ education background.

Highest Education Level	Analogy-I	Analogy-II	Classification	Synonym	Antonym	GRE-Antonym
High school	57	81	68	87	63	47
Bachelor’s degree or candidate	109	67	96	70	98	96
Master’s degree or candidate	26	47	24	33	30	46
Doctorate degree or candidate	8	5	12	10	9	11
Overall	200	200	200	200	200	200

five Mechanical Turk jobs, one job corresponding to one specific question type. The jobs were delivered to 200 people. During this study, we collected the information of the participants’ age and education background, and also measured the accuracy with respect to different age range or education background, separately.

Latent Dirichlet Allocation Model (LDA). This baseline model leveraged one of the most classical distributional word representations, i.e. Latent Dirichlet Allocation (LDA) [5]. In particular, we trained word representations using LDA on *wiki2014* with the topic number 1000.

Skip-Gram Model (SG). In this baseline, we applied the word embedding trained by skip-gram [21] (denoted by **SG-1**). In particular, when using skip-gram to learn the embedding on *wiki2014*, we set the window size as 5, the embedding dimension as 500 and 1000, the negative sampling count as 3, and the epoch number as 3. In addition, we also employed a pre-trained word embedding by Google⁶ with the dimension of 300 (denoted by **SG-2**).

Multi-Sense Model (MS). In this baseline, we applied the multi-sense word embedding models proposed in [17] and [31] (denoted by **MS-1** and **MS-2** respectively). For **MS-1**, we directly used the published multi-sense word embedding vectors by the authors⁷, in which they set 10 senses for the top 5% most frequent words. For **MS-2**, we adopted the same configurations as **MS-1**.

Relation Knowledge Powered Model (RK). This is our proposed method in Section 4. In particular, when learning the embedding on *wiki2014*, we set the window size as 5, the embedding dimension as 500, the negative sampling count as 3 (i.e. the number of random selected negative triples in S'), and the epoch number as 3. We adopted the online Longman Dictionary as the dictionary used in multi-sense clustering. We used a public relation knowledge set, WordRep [15], for relation training.⁸

5.3 Experimental Results

5.3.1 Accuracy of Question Classifier

We applied the question classifier trained in Section 5.1.2 on the test set in Table 1, and got the total accuracy 93.1%.

⁶<https://code.google.com/p/word2vec/>

⁷<http://ai.stanford.edu/~ehhuang/>

⁸Note that Sanghi et al. [26] built an IQ test solver using WWW-based computer program. However, they only introduced the solution to the Classification question but did not discuss the solutions to the other types of verbal questions. Thus, we will not compare our model with this approach.

Table 5: Accuracy of GRE Antonym Question.

	GRE-Antonym
RG	20.60
HP	
18-29	51.17
30-39	53.18
40-60	61.98
Over 60	64.62
High school	51.06
Bachelor’s degree or candidate	55.82
Master’s degree or candidate	60.12
Doctorate degree or candidate	67.19
Overall	56.32
SG	
SG-1	41.97
SG-2	45.68
MS	
MS-1	39.50
MS-2	37.65
RK	52.46

In RG and HP, the question classifier was not needed. In SG and RK, the wrongly classified questions were also sent to the corresponding wrong solver to find an answer. If the solver returned an empty result (which was usually caused by invalid input format, e.g., an Analogy-II question was wrongly input to the Classification solver), we would randomly select an answer.

5.3.2 Participants for Measuring Human Performance

We delivered the test questions to human beings through Amazon Mechanical Turk to collect human answers on the test questions. To gain a better understanding on the performance of human, we summarized the statistics of the participants to measure the human performance.

Table 2 shows the statistics of the distribution of participants’ age over each type of test questions. From this table, we can find that most of the participants are in the age of 18-39 for every type of test questions. Some specific question types, such as Classification and Antonym, tend to attract more younger participants compared with elder ones.

Table 3 reports the statistics of the distribution of participants’ education background over each type of test questions. From this table, we can observe that, for every test question type, more than 80% participants hold either high school or bachelor as their highest education levels, while the others hold even higher education degrees. Such statistics ensure that our participants might represent the normal human intelligence.

Table 4: Accuracy of different methods among different human groups.

	Analogy-I	Analogy-II	Classification	Synonym	Antonym	Total
RG	24.60	11.72	20.75	19.27	23.13	20.51
HP						
18-29	44.48	31.72	42.85	42.86	49.72	42.33
30-39	44.97	33.27	46.17	52.65	57.90	46.99
40-60	50.00	38.45	52.87	60.73	56.47	51.70
Over 60	49.00	46.55	56.13	72.04	58.62	56.47
High school	44.15	27.87	42.73	44.47	46.78	41.20
Bachelor's degree or candidate	46.64	33.62	48.61	51.23	54.81	46.98
Master's degree or candidate	47.83	48.77	51.72	61.90	61.08	54.26
Doctorate degree or candidate	55.33	37.93	58.49	71.77	70.69	58.84
Overall	45.87	34.37	47.23	50.38	53.30	46.23
SG						
SG-1	38.00	24.14	37.74	45.10	40.82	38.36
SG-2	38.00	20.69	39.62	47.06	44.90	39.66
MS						
MS-1	36.36	19.05	41.30	50.00	36.59	38.67
MS-2	40.00	20.69	41.51	49.02	40.82	40.09
RK	48.00	34.48	52.83	60.78	51.02	50.86

5.3.3 Overall Accuracy

Table 4 demonstrates the accuracy of answering verbal questions by using all the approaches mentioned in Section 5.2. From this table, we can find that our RK model can achieve the best overall accuracy than all the other methods. In particular, our model can raise the overall accuracy by about 6% over HP. We can also observe that, even without using any extra knowledge but context co-occurrence, the SG and MS models can significantly outperform RG. These results are quite impressive, indicating the great potential of using machine to comprehend human knowledge and even achieve the comparable level of human intelligence. In addition, we can observe that our RK model is empirically superior than the two multi-sense algorithms MS-1 and MS-2, demonstrating that it is important to adopt less model parameters and use online dictionary in building the multi-sense embedding model.

5.3.4 Accuracy in Different Question Types

Table 4 also reports the accuracy of answering various types of verbal questions by each comparing method. From the table, we can observe that the SG and MS models can achieve competitive accuracy on some certain question types (like Synonym) compared with HP. After incorporating knowledge into learning word embedding, our RK model can improve the accuracy over all question types. Moreover, the table shows that our RK model can result in a big improvement over HP on the question types of Synonym and Classification, while its accuracy on the other question types is not so significant as these two types.

5.3.5 Comparison with Different Human Age

In addition to the comparison in terms of overall accuracy, Table 4 illustrates the accuracy of answering verbal questions by human with different age segments. From the table, we can find that elder people tend to achieve better overall accuracy than younger groups, while such trend may not be consistent under some certain question types like Antonym. This table also reveals that our RK model can reach the competitive performance of the involved Amazon Mechanical Turk workers under the age from 40 to 60 in the verbal questions, which indicates the potential of the word embedding to comprehend human knowledge and form up

certain intelligence.

5.3.6 Comparison with Different Education Background

Table 4 also compares the accuracy of answering verbal questions by human with different education background. From the table, we can find that people with higher education degrees tend to achieve better accuracy in terms of any question type than those with lower degrees. This is consistent to the common sense. This table also reveals that our RK model can reach the competitive performance between the involved Amazon Mechanical Turk workers with the bachelor degrees and those with the master degrees in the verbal questions, which also implies the potential of the word embedding to comprehend human knowledge and form up certain intelligence.

5.3.7 Accuracy in GRE Antonym Question

Table 5 demonstrates the accuracy of solving GRE Antonym questions by using all the approaches mentioned in Section 5.2. This table shows that our RK model performs better than the RG, SG, and MS models. Also, RK can achieve better accuracy than the human under the age from 18 to 29 and people with high school degree, though not achieve a better performance than the human overall accuracy. This result further demonstrates the effectiveness of our RK method since GRE is assumed to be much more difficult than the verbal questions in standard IQ test.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we investigated how to automatically solve verbal comprehension questions in the Intelligence Quotient (IQ) Test by using AI technologies, especially the deep learning techniques that are recently developed and successfully applied in text mining and natural language processing. To fulfill the challenging task, especially in terms of the multiple senses of words and the complex relations among words, we proposed a novel framework consisting of three components: (i) the first component is a classifier that aims to recognize the specific type of a verbal comprehension question; (ii) the second component leverages a novel deep learning technique to co-learn the representations of both word-sense pairs and relations among words (or their senses); (iii) the last component is comprised of dedicated solvers, based on

the obtained word-sense pair representations and relation representations, for addressing each of the specific types of questions. Experimental results have illustrated that this novel framework can achieve better performance than existing methods for solving verbal comprehension questions and even exceed the average performance of the Amazon Mechanical Turk workers involved in the experiments.

While this work is a very early attempt to solve IQ Test using AI techniques, the evaluation results are highly encouraging and indicate that, with appropriately leveraging the deep learning technologies, we could be a further small step closer to the human intelligence. In the future, we plan to leverage more types of knowledge from the knowledge graph, such as Freebase⁹, to enhance the power of obtaining word-sense and relation embeddings. Moreover, we will explore new frameworks based on deep learning or other AI techniques to solve other parts of IQ tests beyond verbal comprehension questions.

7. REFERENCES

- [1] A. Aizawa. An information-theoretic perspective of tf-idf measures. *Information Processing & Management*, 39(1):45–65, 2003.
- [2] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.
- [3] J. Bian, B. Gao, and T.-Y. Liu. Knowledge-powered deep learning for word embedding. In *Proceedings of ECML/PKDD*, 2014.
- [4] C. M. Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.
- [5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
- [6] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.
- [7] A. Bordes, J. Weston, R. Collobert, Y. Bengio, et al. Learning structured embeddings of knowledge bases. In *AAAI*, 2011.
- [8] P. Carter. *The complete book of intelligence tests*. John Wiley & Sons Ltd, 2005.
- [9] P. Carter. *The Ultimate IQ Test Book: 1,000 Practice Test Questions to Boost Your Brain Power*. Kogan Page Publishers, 2007.
- [10] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*, pages 160–167. ACM, 2008.
- [11] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.
- [12] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [13] D. L. Dowe and J. Hernández-Orallo. Iq tests are not for machines, yet. *Intelligence*, 2012.
- [14] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *Proceedings of SIGCHI*, pages 281–285. ACM, 1988.
- [15] B. Gao, J. Bian, and T.-Y. Liu. Wordrep: A benchmark for research on learning word representations. *arXiv preprint arXiv:1407.1640*, 2014.
- [16] M. J. Hosseini, H. Hajishirzi, O. Etzioni, and N. Kushman. Learning to solve arithmetic word problems with verb categorization. 2014.
- [17] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of ACL*, pages 873–882, 2012.
- [18] P. C. Ken Russell. *The Times Book of IQ Tests*. Kogan Page Limited, 2002.
- [19] N. Kushman, Y. Artzi, L. Zettlemoyer, and R. Barzilay. Learning to automatically solve algebra word problems. In *Proceedings of the conference of the Association for Computational Linguistics*, 2014.
- [20] M.-T. Luong, R. Socher, and C. D. Manning. Better word representations with recursive neural networks for morphology. *CoNLL-2013*, 104, 2013.
- [21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.
- [22] S. Mohammad, B. Dorr, and G. Hirst. Computing word-pair antonymy. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 982–991. Association for Computational Linguistics, 2008.
- [23] D. Pape. *The Original Cambridge Self Scoring IQ Test*. The Magni Group, Inc, 1993.
- [24] B. Pintér, G. Vörös, Z. Szabó, and A. Lörincz. Automated word puzzle generation via topic dictionaries. *CoRR*, abs/1206.0377, 2012.
- [25] J. Reisinger and R. J. Mooney. Multi-prototype vector-space models of word meaning. In *Proc. of HLT*, 2010.
- [26] P. Sanghi and D. Dowe. A computer program capable of passing i.q. tests. In *Proceedings of the Joint International Conference on Cognitive Science*, 2003.
- [27] M. J. Seo, H. Hajishirzi, A. Farhadi, and O. Etzioni. Diagram understanding in geometry questions. 2014.
- [28] R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*, pages 926–934, 2013.
- [29] W. Stern. *The Psychological Methods of Testing Intelligence*. Warwick & York, 1914.
- [30] C. Strannegard, M. Amirghasemi, and S. Ulfbacker. An anthropomorphic method for number sequence problems. *Cognitive Systems Research*, 2012.
- [31] F. Tian, H. Dai, J. Bian, B. Gao, R. Zhang, E. Chen, and T.-Y. Liu. A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of the 25th International Conference on Computational Linguistics*, 2014.
- [32] A. M. Turing. Computing machinery and intelligence. *Mind*, 49:433–460, 1950.
- [33] D. Wechsler. Wechsler adult intelligence scale—fourth edition (wais-iv). *San Antonio, TX: NCS Pearson*, 2008.
- [34] J. Weston, A. Bordes, O. Yakhnenko, and N. Usunier. Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973*, 2013.
- [35] R. W. Woodcock, K. S. McGrew, and N. Mather. *Woodcock-Johnson III tests of cognitive abilities*. Riverside Pub., 2001.
- [36] C. Xu, Y. Bai, J. Bian, B. Gao, G. Wang, X. Liu, and T.-Y. Liu. Rc-net: A general framework for incorporating knowledge into word representations. In *CIKM’14*.
- [37] M. Yu and M. Dredze. Improving lexical embeddings with semantic knowledge. In *Association for Computational Linguistics (ACL)*, 2014.

⁹<http://www.freebase.com/>